# 微服务开发文档(javahub 网站举例)

---王亮

本demo（javahub 网站）是基于jcloudspace 框架开发的微服务项目，jcloudspace 是我2018年前后开发连锁药房项目时候沉淀整理的微服务框架(基于spring cloud )，后期利用业余时间进行了升级，前端是基于vue(element ui)开发的，起名:jwebframework。

前端框架（jwebframework）：

封装了登录，退出，页面tab切换，权限拦截，异常拦截，消息提示，自动分页，用户管理，角色管理，权限组管理，权限管理，黑白名单管理，定时任务管理，sql监控，cpu监控，性能曲线监控等页面和js 通讯共通，开发人员只要直接开发业务页面就可以了。

后台框架（jcloudspace ）：

封装了权限管理，登录退出，sql监控，页面配置定时任务，elk日志处理，redis操作，stream kafka 通信，数据库动态读写分离（自己写的一套可扩展的数据库管理接口，目前只支持mysql，实现接口就可以集成不同的数据库），支持model 直接增删改查，类似7.0的对model的操作，elasticserch 操作，跨库查询，tx-lnc分布式事务管理，Zull gateway网关等，开发人员可以直接新集成一个业务项目进来就可以直接运行了。

Javahub 网站（正在备案 域名：javahub.top）：

Javahub是类似csdn的一个知识分享网站，目前还没发布上线，下班后业余时间正在慢慢改进，预计五一左右可以发布上线，这个网站是为了这个框架的应用开发的一个demo,不久后框架1.0版和javahub都讲在github开源，目的是讲具体业务集成进框架来展示和测试框架的易用性和扩展性。

Jcloudspace 框架整体架构图

JZUUL

EUREKA-ZUUl/GATEAWAY(所有请求网关入口)

（并负责所有请求的权限校验和请求过滤）

JSENTER

EUREKA-SERVER（EUREKA 注册中心，负责管理所有服务）

# 

JPARENT

PARENT-CPDE(每个项目都需要maven引入这个项目，负责共通类的引用)

JSETTING

EUREKA-CLIENT（负责用户权限，权限集合，

角色，定时任务，性能监控，sql日志等系统设置部分的后台服务）

JSERVICE

EUREKA-CLIENT(负责所有共通api调用服务，比如登录,退出等)

JWIKI

EUREKA-CLIENT（负责所有系统的日志服务，和数据记录，例如启动了这个服务，系统就会增加了一个wiki百科管理的功能）

JBASEINFO

EUREKA-CLIENT（负责例如系统用户信息管理等基本信息的后台服务）

JPUB（应用demo）

EUREKA-CLIENT（如果启动了这个就可以增加类似csdn的一个业务应用具体内容下面会介绍）

JAVAHUB：

JWEBFRAMEWORK 框架整体架构图

node js（Webpack ）容器

VUE(ELEMENT-UI)

## 项目启动依赖服务

Redis

Zookeeper

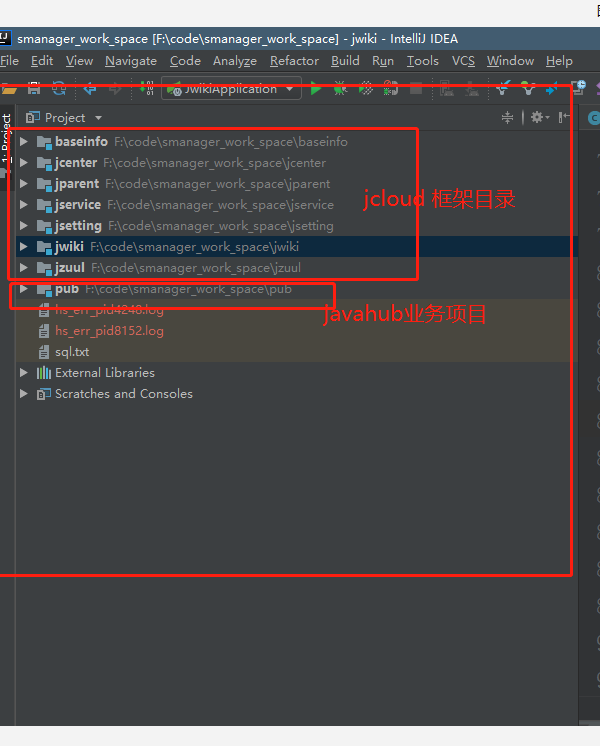
Kafka

Mysql

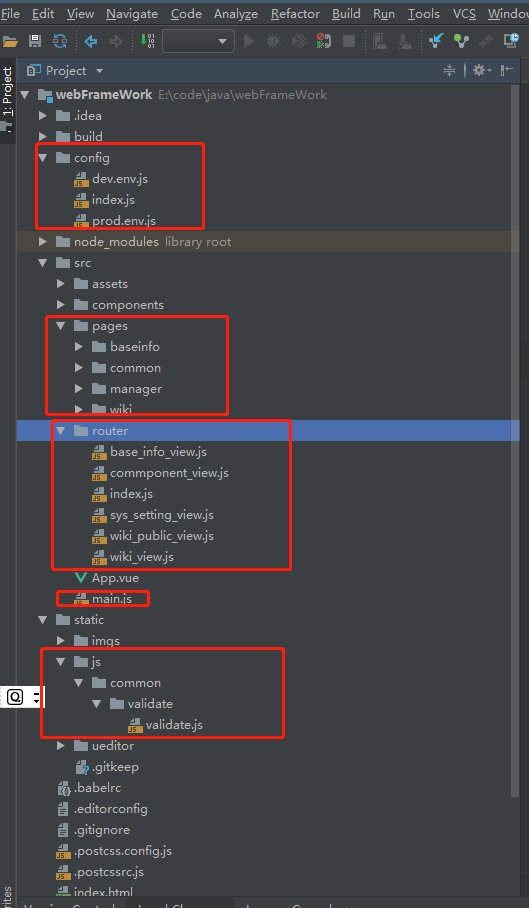
Node.js

Elasticserch

Jcloudspace 目录结构



JWEBFRAMEWORK 目录结构



每个文件夹作用和每个文件如何配置将在后续介绍

未完待续（后面内容更精彩，会发布在github上不定期更新）

1.什么是微服务

2.为什么微服务选择spring cloud

3.spring cloud 运行原理是什么

4.什么是spring cloud 注册中心和模块注册以及如何实现

5.什么是spring cloud 网关和高可用以及如何实现

6.什么是spring cloud 负载均衡和feign 和ribbon 服务间调用以及如何实现

7.什么是spring cloud 熔断和服务降级

8.怎么做到spring cloud 事物一致性

9.什么是spring cloud 配置中心以及如何实现

10.什么是spring cloud even总线以及如何实现

11.什么是spring cloud stream 消息总线以及rabbit 和kafka如何实现

12.如何实现spring cloud 多服务下日志链路追踪以及如何实现

13.结合jcloudspace和javahub 项目具体说明以上每一部分在项目中如何应用

注：因为框架是从连锁药房系统摘出来的，所以未经大范围修改（正在用业余时间修改，预计五一左右可能开源发布，具体github地址未定）禁止用于商业项目（开源后的从我的github down下来的代码用于商业项目需遵循开源协议），防止不必要的商业纠纷，可用于学习微服务用框架